Έγγραφο απαιτήσεων λογισμικού (SRS)

ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ ΑΝΤΙΣΤΟΙΧΟΥ ΕΓΓΡΑΦΟΥ ΤΟΥ ΠΡΟΤΥΠΟΥ ISO/IEC/IEEE 29148:2011

Σύστημα Διαμοιρασμού Δεδομένων Κατανάλωσης & Παραγωγής Ηλεκτρικής Ενέργειας

# Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

Στόχος της ανάπτυξης του συγκεκριμένου συστήματος είναι η εύκολη και άμεση πρόσβαση των ενδιαφερόμενων χρηστών σε δεδομένα που αφορούν την κατανάλωση ηλεκτρικής ενέργειας σε όλες τις ενεργειακές ζώνες της Ευρώπης. Οι χρήστες είναι σε θέση να γνωρίζουν, εκτός από παρελθοντικά δεδομένα κατανάλωσης ενέργειας, προβλέψεις για τα επίπεδα στα οποία θα κυμανθεί η κατανάλωση σε μελλοντικά χρονικά διαστήματα (συγκεκριμένη μέρα, συγκεκριμένο μήνα ή συγκεκριμένη χρονιά). Επιπλέον, είναι διαθέσιμα δεδομένα σχετικά με τους διάφορους τρόπους παραγωγής ηλεκτρικής ενέργειας και το ποσό ηλεκτρικής ενέργειας το οποίο συνεισφέρει η κάθε μέθοδος παραγωγής στο συνολικό φορτίο. Η εύκολη διανομή αυτών των δεδομένων, συμβάλλει στην καλύτερη οργάνωση παραγωγής και διαχείρισης ενέργειας, έτσι ώστε να μην παράγονται μεγάλα ενεργειακά πλεονάσματα που μένουν ανεκμετάλλευτα, αλλά και να μην διατίθενται ενέργεια λιγότερη από τη ζητούμενη, καθώς η εκτός προγράμματος παραγωγή ηλεκτρικής ενέργειας είναι εξαιρετικά κοστοβόρα διαδικασία.

## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

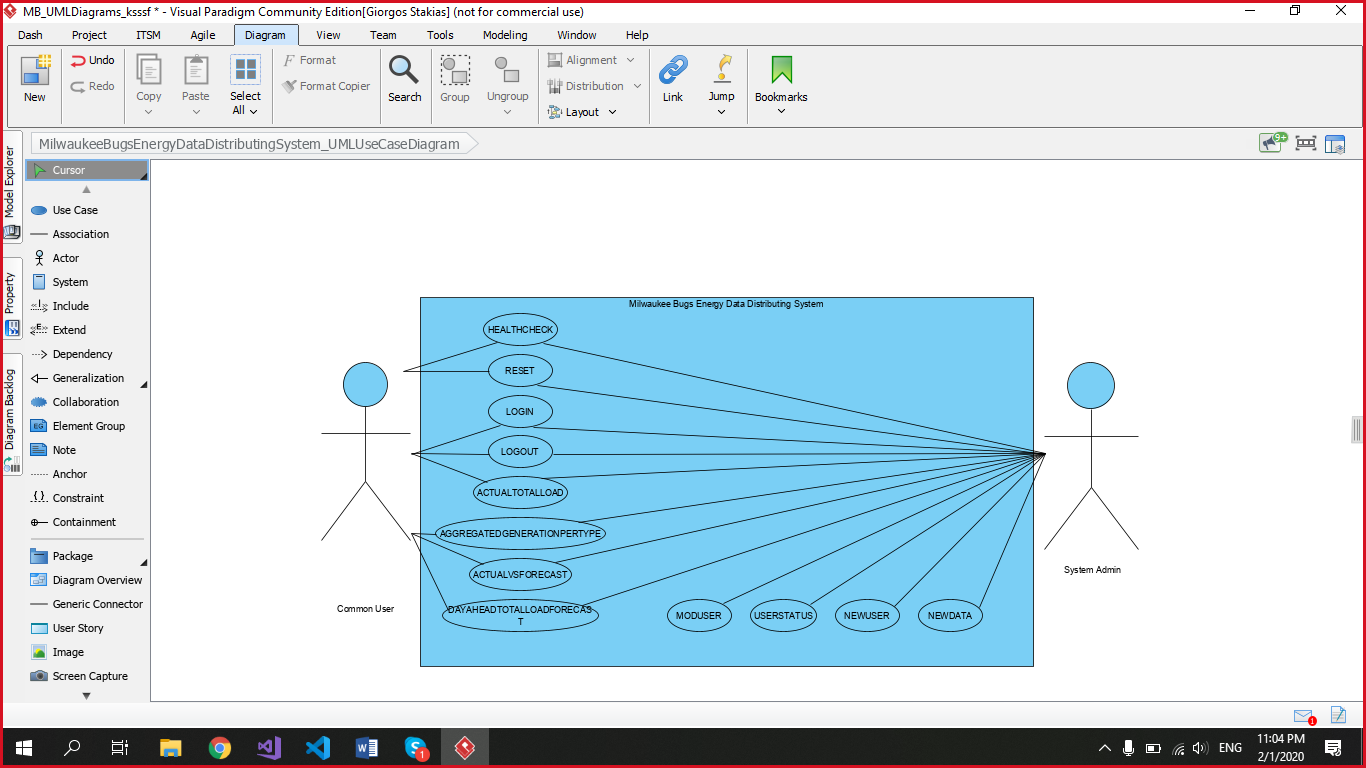
Η δεδομένη πλατφόρμα αξιοποιεί ως εξωτερικό σύστημα τον διεθνή οργανισμό του European Networks Transmission Systems Operators for Electricity (Entsoe), από το οποίο ενημερώνει τις εγγραφές τις βάσης με τα νέα δεδομένα των μετρούμενων μεγεθών. Στη παρούσα υλοποίηση και για ακαδημαϊκούς σκοπούς νέα δεδομένα μπορούν να προστεθούν στη βάση μόνο από τον διαχειριστή του συστήματος χειροκίνητα.

Εφόσον, στη παρούσα υλοποίηση δεν υπάρχει αυτοματοποιημένη σύνδεση – επικοινωνία με τον παραπάνω οργανισμό, δεν ορίζεται και κάποια διεπαφή. Ωστόσο, γενικά οποιοδήποτε εξωτερικό σύστημα επιχειρεί να συνδεθεί με το σύστημά μας, αξιοποιεί τη REST API διεπαφή που διαθέτουμε.

### 1.2.2 Διεπαφές με το χρήστη

Οι διάφοροι χρήστες της εφαρμογής έχουν τη δυνατότητα πρόσβασης στα διανεμόμενα δεδομένα μέσω τριών διαφορετικών διεπαφών χρήστη: ένα Command Line Interface, ένα Web Application και ένα Mobile Application. Αμφότερες οι παραπάνω διεπαφές διακρίνουν δύο είδη χρηστών, τον διαχειριστή του συστήματος (admin) και τον κοινό χρήστη (common user, με μειωμένα δικαιώματα) στους οποίους παρέχουν τις εξής δυνατότητες:

* Κοινές δυνατότητες:
  + HealthCheck: λειτουργία με την οποία ελέγχεται εύρυθμη λειτουργία των back end server και database server καθώς επίσης και η δυνατότητα σύνδεσης οποιασδήποτε διεπαφής. Για τη συγκεκριμένη λειτουργία δεν απαιτείται κανένα επίπεδο διαπίστευσης χρήστη.
  + Reset: λειτουργία η οποία διαγράφει από τη βάση δεδομένων όλους τους υπάρχοντες χρήστες, εκτός του διαχειριστή συστήματος, καθώς επίσης και όλες τις εγγραφές στους πίνακες ActualTotalLoad, AggregatedGenerationPerType και DayAheadTotalLoadForecast. Για τη συγκεκριμένη λειτουργία δεν απαιτείται κανένα επίπεδο διαπίστευσης χρήστη.
  + Login: λειτουργία μέσω της οποίας γίνεται διαπίστευση χρήστη και απόδοση ενός token στον εκάστοτε χρήστη. Για ΟΛΕΣ τις υπόλοιπες δυνατότητες που παρέχονται από τις διεπαφές χρήστη, ο εκάστοτε χρήστης πρέπει να είναι διαπιστευμένος. Για τη συγκεκριμένη λειτουργία δεν απαιτείται κανένα επίπεδο διαπίστευσης χρήστη, ενώ ο χρήστης οφείλει να εισάγει το μοναδικό συνδυασμό username/password που διαθέτει.
  + Logout: λειτουργία με την οποία ο χρήστης τερματίζει τη σύνδεση του με το σύστημα. Παράλληλα διαγράφεται το αρχείο που δημιουργήθηκε κατά τη διαδικασία του Login και περιέχει το token του χρήστη.
  + ActualTotalLoad: λειτουργία με την οποία ο χρήστης δύναται να επιλέξει μία περιοχή και μία συγκεκριμένη χρονική περίοδο με την επιθυμητή χρονική ανάλυση, ώστε να του επιστραφεί το αντίστοιχο σύνολο δεδομένων, που αφορούν την πραγματική κατανάλωση ενέργειας της επιλεγμένης περιοχής για την επιλεγμένη χρονική περίοδο. Η προαναφερθείσα μπορεί να αναφέρεται είτε σε μία συγκεκριμένη μέρα, είτε έναν συγκεκριμένο μήνα, είτε και σε ένα ολόκληρο χρόνο με την προϋπόθεση να έχει επέλθει. Όσον αφορά την οπτικοποίηση των δεδομένων αυτών, το CLI προσφέρει μία υποτυπώδη, στη μορφοποίηση που αιτήθηκε ο χρήστης (json ή csv), ενώ το WEB και Mobile Application παρέχουν μία πιο φιλική παρουσίαση με τη χρήση διαγραμμάτων και πινάκων. Επιπλέον, παρέχεται η δυνατότητα λήψης του εκάστοτε αρχείου.
  + AggregatedGenerationPerType: λειτουργία με την οποία ο χρήστης δύναται να επιλέξει μία περιοχή, μία χρονική περίοδο, με την επιθυμητή χρονική ανάλυση και έναν τύπο παραγωγής ενέργειας (π.χ. πυρηνική), ώστε να του επιστραφεί το αντίστοιχο σύνολο δεδομένων, που αφορούν την παραγωγή ενέργειας, από το συγκεκριμένο τύπο, της επιλεγμένης περιοχής για την επιλεγμένη χρονική περίοδο. Η προαναφερθείσα μπορεί να αναφέρεται είτε σε μία συγκεκριμένη μέρα, είτε έναν συγκεκριμένο μήνα, είτε και σε ένα ολόκληρο χρόνο. Η οπτικοποίηση των δεδομένων ακολουθεί τα πρότυπα της ActualTotalLoad λειτουργίας.
  + DayAheadTotalLoadForecast: λειτουργία με την οποία ο χρήστης δύναται να επιλέξει μία περιοχή και μία συγκεκριμένη χρονική περίοδο με την επιθυμητή χρονική ανάλυση, ώστε να του επιστραφεί το αντίστοιχο σύνολο δεδομένων, που αφορούν την πρόβλεψη κατανάλωσης ηλεκτρικής ενέργειας της επιλεγμένης περιοχής για την επιλεγμένη χρονική περίοδο. Στη συγκεκριμένη λειτουργία, η χρονική περίοδος μπορεί να αφορά την τρέχουσα, κάποια παρελθοντική και ακόμη κάποια μελλοντική ημερομηνία, σε αντίθεση με τις δύο προηγούμενες λειτουργίες. Η οπτικοποίηση των δεδομένων ακολουθεί τα πρότυπα της. ActualTotalLoad λειτουργίας.
  + ActualvsForecast: λειτουργία με την οποία ο χρήστης δύναται να επιλέξει μία περιοχή και μία συγκεκριμένη χρονική περίοδο με την επιθυμητή χρονική ανάλυση, ώστε να του επιστραφεί το αντίστοιχο σύνολο δεδομένων, που αφορούν τόσο την πραγματική κατανάλωση ηλεκτρικής ενέργειας, όσο και την αντίστοιχη πρόβλεψή της, για την επιλεγμένη περιοχή και για την επιλεγμένη χρονική περίοδο. Η οπτικοποίηση των δεδομένων ακολουθεί τα πρότυπα της ActualTotalLoad λειτουργίας, με το επιπλέον χαρακτηριστικό της σύγκρισης των εν λόγω μεγεθών αναπαριστώντας αυτά σε κοινούς πίνακες και γραφήματα (WEB Application & Mobile Application).
* Επιπλέον λειτουργίες του διαχειριστή συστήματος:
  + Newuser: λειτουργία με την οποία ο διαχειριστής συστήματος δημιουργεί έναν νέο χρήστη και τον εισάγει στη βάση δεδομένων.
  + Moduser: λειτουργία με την οποία ο διαχειριστής του συστήματος τροποποιεί τα στοιχεία του χρήστη.
  + Userstatus: λειτουργία με την οποία ο διαχειριστής του συστήματος μπορεί να δει την κατάσταση ενός χρήστη.
  + Newdata: λειτουργία με την οποία ο διαχειριστής του συστήματος μπορεί να εισάγει στη βάση ένα νέο σύνολο δεδομένων μέσω ενός αρχείου csv.



# Αναφορές - πηγές πληροφοριών

Αναφορά σε πηγές πληροφοριών στο μέτρο της αναγκαιότητας για την κατανόηση του συστήματος

ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΙ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

#### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Σύνδεση χρήστη - Login

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Η συγκεκριμένη περίπτωση χρήσης αφορά τόσο τους κοινούς χρήστες όσο και τον διαχειριστή συστήματος.

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

Δεν υπάρχει καμία προϋπόθεση για την εκτέλεση της συγκεκριμένης περίπτωσης, καθώς οποιοσδήποτε μπορεί να αποπειραθεί να συνδεθεί στο σύστημα. Προφανώς, χρήστες που δεν είναι καταγεγραμμένοι στη βάση δεν γίνονται αποδεκτοί.

#### 3.1.1.3 Περιβάλλον εκτέλεσης

Οποιαδήποτε διεπαφή από τις τρεις πιθανές υποστηρίζει τη λειτουργία σύνδεσης χρήστη. Ταυτόχρονα, η διαδικασία εμπλέκει το περιβάλλον του backend και το DBMS.

#### 3.1.1.4 Δεδομένα εισόδου

Δεδομένα εισόδου δεν υπάρχουν στη συγκεκριμένη περίπτωση χρήσης. Με την επιτυχή ολοκλήρωση της λειτουργίας αυτής επιστρέφεται ένα token, το οποίο και αποθηκεύεται σε ένα αρχείο στη μεριά του πελάτη.

#### 3.1.1.5 Παράμετροι

Για την ολοκλήρωση της εκτέλεσης απαιτείται η συμπλήρωση ενός μοναδικού συνδυασμού username και password.

#### 3.1.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Η διαδικασία εκτέλεσης της λειτουργίας αυτής είναι η ακόλουθη:

* Βήμα 1. Έλεγχος αν ο χρήστης είναι ήδη συνδεδεμένος. Αν αυτό ήταν αληθές, τότε η διαδικασία ολοκληρώνεται χωρίς κάποιο επιπλέον βήμα. Εναλλακτικά, μεταφερόμαστε στο Βήμα 2.
* Βήμα 2. Εισαγωγή των username και password.
* Βήμα 3. Δημιουργία ενός REST API αιτήματος με τα διαπιστευτήρια του χρήστη και αποστολή στο back end του συστήματος.
* Βήμα 4. Αναζήτηση username στη βάση. Εάν αυτό βρεθεί μεταφερόμαστε στο επόμενο βήμα, ενώ σε αντίθετη περίπτωση μεταφερόμαστε στο Βήμα 6.
* Βήμα 5. Επαλήθευση του password που δόθηκε. Στην περίπτωση που αυτό είναι σωστό, ακολουθεί το Βήμα 7, αλλιώς ακολουθεί το Βήμα 6.
* Βήμα 6. Εμφάνιση μηνύματος για το σφάλμα που προέκυψε κατά τη διαδικασία σύνδεσης και μεταφορά στο Βήμα 8.
* Βήμα 7. Δημιουργία αρχείου token του χρήστη και ολοκλήρωση της διαδικασίας.
* Βήμα 8. Δυνατότητα του χρήστη του επανάληψη εισόδου των διαπιστευτηρίων και νέα προσπάθεια σύνδεσης. Παράλληλα δίνεται η δυνατότητα αγνόησης της παραπάνω επιλογής και ολοκλήρωση της διαδικασίας. Εάν ο χρήστης αξιοποιήσει τη δυνατότητα αυτή, μεταφερόμαστε στο Βήμα 2.

Σχετικά UML διαγράμματα δραστηριοτήτων και αλληλουχίας παραδίδονται στο σχετικό .vpp αρχείο.

#### 3.1.1.7 Δεδομένα εξόδου

Το μοναδικό παράγωγο της όλης διαδικασίας είναι το token του χρήστη, που προκύπτει μετά από την επιτυχή του σύνδεση.

### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: ActualTotalLoad

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Η συγκεκριμένη περίπτωση χρήσης αφορά τόσο τους κοινούς χρήστες όσο και τον διαχειριστή συστήματος.

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

Για την εκτέλεση αυτής της περίπτωσης, απαιτείται ο χρήστης να είναι συνδεδεμένος στο σύστημα και το αντίστοιχο πεδίο quota να είναι μεγαλύτερο του μηδενός.

#### 3.1.2.3 Περιβάλλον εκτέλεσης

Οποιαδήποτε διεπαφή από τις τρεις πιθανές υποστηρίζει τη λειτουργία ActualTotalLoad με κάποια σχετική ή όχι οπτικοποίηση. Ταυτόχρονα, η διαδικασία εμπλέκει προφανώς το περιβάλλον του backend και το DBMS.

#### 3.1.2.4 Δεδομένα εισόδου

Δεδομένα εισόδου δεν υπάρχουν στην συγκεκριμένη περίπτωση χρήσης. Η επιτυχής εκτέλεση επιστρέφει μία λίστα με εγγραφές της βάσης, οι οποίες οπτικοποιούνται από τις διεπαφές χρήστη. Αυτή είναι και η έξοδος της παραπάνω περίπτωσης χρήσης.

#### 3.1.2.5 Παράμετροι

Απαιτείται η συμπλήρωση των εξής πεδίων:

* Area: Το όνομα της περιοχής για την οποία ο χρήστης αιτείται τα δεδομένα.
* TimeRes: Η χρονική ανάλυση (PT60M / PT30M / PT15M).
* Date: Ημερομηνία με κάποια μορφοποίηση (YYYY-MM-DD / YYYY-MM / YYYY).

Ο χρήστης μπορεί προσδιορίσει (προαιρετικά) και τη μορφοποίηση της εξόδου (json / csv). Εάν δεν αποδώσει τιμή στο πεδίο format, η προεπιλεγμένη μορφοποίηση είναι json.

Συνθήκες εγκυρότητας των παραμέτρων είναι οι εξής:

* Ο χρήστης συμπληρώνει όλα τα υποχρεωτικά πεδία.
* Οι τιμές που δίνει ο χρήστης για το πεδίο timeres είναι μία εκ των {PT60M / PT30M / PT15M}.
* Η δοθείσα ημερομηνία ακολουθεί κάποια από τις αποδεκτές μορφοποιήσεις.
* Αν ο χρήστης δώσει τιμή για το πεδίο format, αυτή πρέπει να είναι μία εκ των {json / csv}.
* Για την συγκεκριμένη λειτουργία, η ημερομηνία απαιτείται να είναι παρελθοντική.

#### 3.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Η διαδικασία εκτέλεσης της λειτουργίας αυτής είναι η ακόλουθη:

* Βήμα 1. Εισαγωγή των area, timers, date και προαιρετικά format.
* Βήμα 2. Έλεγχος για το αν ο χρήστης είναι συνδεδεμένος, οι παράμετροι εισόδου είναι σωστά μορφοποιημένες και αν ο χρήστης έχει εναπομείναντα quotas. Αν κάποια από τις τρεις συνθήκες δεν ισχύει μεταφερόμαστε στο Βήμα 6.
* Βήμα 3. Δημιουργία ενός REST API αιτήματος με τις δοσμένες παραμέτρους και αποστολή στο back end του συστήματος.
* Βήμα 4. Έλεγχος απάντησης της βάσης. Εάν δεν βρέθηκαν εγγραφές, μεταφερόμαστε στο Βήμα 6.
* Βήμα 5. Οπτικοποίηση των αποτελεσμάτων της βάσης και ολοκλήρωση της διαδικασίας.
* Βήμα 6. Εμφάνιση μηνύματος εξαίρεσης / σφάλματος και ολοκλήρωση της διαδικασίας.

Σχετικά UML διαγράμματα δραστηριοτήτων και αλληλουχίας παραδίδονται στο σχετικό .vpp αρχείο.

#### 3.1.2.7 Δεδομένα εξόδου

Το παράγωγο της όλης διαδικασίας είναι η οπτικοποίηση των σχετικών εγγραφών της βάσης στο standard output για το CLI και σε κάποια φόρμα (πίνακας ή και γράφημα) για το Web και Mobile Application, καθώς επίσης και τα διαθέσιμα προς λήψη αρχεία.

#### 3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: HealthCheck

#### 3.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται

Η συγκεκριμένη περίπτωση χρήσης αφορά τόσο τους κοινούς χρήστες όσο και τον διαχειριστή συστήματος.

#### 3.1.3.2 Προϋποθέσεις εκτέλεσης

Δεν υπάρχει καμία προϋπόθεση για την εκτέλεση της συγκεκριμένης περίπτωσης, καθώς οποιοσδήποτε μπορεί να εκτελέσει την περίπτωση χρήσης HealthCheck ώστε να διαπιστώσει εάν ο back-end-server και o db-server είναι ενεργοί.

#### 3.1.3.3 Περιβάλλον εκτέλεσης

Οποιαδήποτε διεπαφή από τις τρεις πιθανές υποστηρίζει την συγκεκριμένη περίπτωση χρήσης. Από το CLI ο ίδιος ο χρήστης θα πρέπει να εκτελεί χειροκίνητα την εντολή ενώ στο Web και το Mobile App θα εκτελείται αυτόματα ο συγκεκριμένος έλεγχος με την εκκίνηση της εφαρμογής. Ταυτόχρονα, η διαδικασία εμπλέκει προφανώς το περιβάλλον του backend και το DBMS.

#### 3.1.3.4 Δεδομένα εισόδου

Δεδομένα εισόδου δεν υπάρχουν στη συγκεκριμένη περίπτωση χρήσης. Σε περίπτωση επιτυχίας του ελέγχου δεν έχουμε δεδομένα εξόδου. Σε περίπτωση αποτυχίας έχουμε την εμφάνιση ενός μηνύματος λάθους.

#### 3.1.3.5 Παράμετροι

Δεν έχει παραμέτρους.

#### 3.1.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Η διαδικασία εκτέλεσης της λειτουργίας αυτής είναι η ακόλουθη:

* Βήμα 1. Δημιουργία ενός REST API αιτήματος με τις δοσμένες παραμέτρους και αποστολή στο back end του συστήματος.
* Βήμα 2. Έλεγχος συνδεσιμότητας με τον εξυπηρετητή του back-end και τη βάση. Εάν αποτύχει η προσπάθεια σύνδεσης μεταφερόμαστε στο Βήμα 4.
* Βήμα 3. Έλεγχος απάντησης της βάσης. Εάν ο κωδικός της http απάντησης δεν είναι 200 μεταφερόμαστε στο Βήμα 4 . Διαφορετικά ολοκληρώνεται η διαδικασία επιτυχώς.
* Βήμα 4. Εμφάνιση μηνύματος εξαίρεσης / σφάλματος και ολοκλήρωση της διαδικασίας.

Σχετικά UML διαγράμματα δραστηριοτήτων και αλληλουχίας παραδίδονται στο σχετικό .vpp αρχείο.

#### 3.1.3.7 Δεδομένα εξόδου

Σε περίπτωση επιτυχίας του ελέγχου δεν έχουμε δεδομένα εξόδου. Σε περίπτωση αποτυχίας έχουμε την εμφάνιση ενός μηνύματος λάθους.

## 3.2 Απαιτήσεις επιδόσεων

Οι επιδόσεις του συστήματος περιστρέφονται γύρω από τέσσερις βασικούς άξονες:

* Τον χρόνο απόκρισης
* Το φόρτο εργασίας που μπορεί να εξυπηρετήσει
* Την κλιμακωσιμότητα

Με σκοπό την ελαχιστοποίηση του χρόνου απόκρισης, η όποια οπτικοποίηση των δεδομένων σε διαγράμματα και πίνακες παράγεται εξ’ ολοκλήρου στη πλευρά του χρήστη. Με αυτόν τον τρόπο, ελαχιστοποιείται και το απαιτούμενο εύρος ζώνης για την αποστολή και λήψη των μηνυμάτων/δεδομένων των ενδιάμεσων συνιστωσών.

Όσον αφορά το δυνατό throughput, για την ακριβή τεκμηρίωση των δυνατοτήτων της εφαρμογής μας απαιτούνται περαιτέρω δοκιμές (benchmarking). Θεωρητικά, επιθυμούμε το σύστημά μας να είναι σε θέση να εξυπηρετήσει αρκετές χιλιάδες αιτήματα ανά χρονική στιγμή.

Τέλος, το σύστημα έχει τη δυνατότητα κλιμάκωσης, τόσο οριζόντιας όσο και κάθετης. Στα πλαίσια της εργασίας, αρκούμαστε στους αρχικά διαθέσιμους πόρους μας.

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Όπως έχει αναφερθεί προηγουμένως, πρόσβαση στα δεδομένα παραγωγής και κατανάλωσης ηλεκτρικής ενέργειας από το σύστημά μας, έχουν μόνο διαπιστευμένοι (συνδεδεμένοι) χρήστες, τους οποίους έχει δημιουργήσει και προσθέσει στη βάση ο διαχειριστής. Επιπροσθέτως, κάθε χρήστης που αιτείται δεδομένα από το σύστημα θα πρέπει να έχει εναπομείναντα quotas για τη συγκεκριμένη περίοδο σύνδεσης.

## 3.4 Περιορισμοί σχεδίασης

Το σύστημά μας εμπεριέχει τις ακόλουθες συνιστώσες:

* Τη βάση δεδομένων
* Το back-end
* Το Command Line Interface (CLI)
* Το Web Application
* Το Mobile Application
* To Rest API, το οποίο σερβίρει δεδομένα στις τρεις παραπάνω διεπαφές
* Το Data Model για την οργάνωση των εγγραφών της βάσεις σε αντικείμενα κλάσεων.

Για τις παραπάνω συνιστώσες χρησιμοποιήθηκαν οι εξής τεχνολογίες - εργαλεία αντίστοιχα:

* MySQL
* Java, Groovy, Spring Framework, JDBC Template
* Java, Picocli Framework
* Javascript, HTML, React Library
* Java, Android Studio
* Java
* Java

Παράλληλα, όλα τα αιτήματα του χρήστη μεταφέρονται στον εξυπηρετητή του back-end με χρήση https πρωτοκόλλου.

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Για την χρήση της πλατφόρμας από την Web εφαρμογή δεν απαιτείται κάποια περαιτέρω εγκατάσταση, καθώς το site θα είναι προσβάσιμο από οποιονδήποτε πλοηγό ιστού (browser). Η Mobile εφαρμογή θα είναι διαθέσιμη σε έκδοση στο Google Playstore με κόστος που εξισορροπεί τα ψυχικά τραύματα της σύνταξης αυτού εγγράφου. Σε περίπτωση δημιουργίας νέας έκδοσης του λογισμικού, θα φροντίσουμε ώστε να μπορούν κατά το δυνατό να τρέχουν και οι παλιότερες εκδόσεις, χωρίς ωστόσο αυτές να είναι διαθέσιμες για λήψη και εγκατάσταση.

### 3.5.2 Απαιτήσεις ασφάλειας

Για την διασφάλιση της προστασίας των προσωπικών δεδομένων των χρηστών, καμία σχετική πληροφορία δεν αποθηκεύεται στη βάση ως plain text, αντιθέτως αξιοποιούνται συναρτήσεις hashing. Με τον τρόπο αυτό ούτε ο ίδιος ο διαχειριστής δεν θα έχει πρόσβαση στα δεδομένα των χρηστών. Παράλληλα, η δραστηριότητα του χρήστη δεν καταγράφεται, ούτε αποθηκεύεται.

Για την προστασία του ίδιου του συστήματος από εξωτερικούς κακόβουλους παράγοντες, οι επικοινωνία των διαφόρων συνιστωσών είναι κρυπτογραφημένη με χρήση του πρωτοκόλλου https. Επιπλέον, σε κάθε αίτημα χρήστη, οι παράμετροι που δίνονται φιλτράρονται για την αποφυγή sql injections.

### 3.5.3 Απαιτήσεις συντήρησης

*Θα δεχόμαστε σχόλια για την εφαρμογή μας από τους χρήστες, με τη βοήθεια των οποίων θα εντοπίζουμε bugs και θα προσπαθούμε να τα διορθώσουμε. Επίσης θα επεξεργαζόμαστε προτάσεις για βελτίωση της εφαρμογής. Αυτοματοποιημένα τεστ στα οποία θα υποβάλλουμε το λογισμικό καθόλα τα στάδια ανάπτυξης και ζωής.*